Attorney Docket No.: 42390.P8628 Patent

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	Carl M. Ellison et al.	)		
		)		
Serial No.	09/540,613	)	Art Unit:	2134
		)		
Filed:	March 31, 2000	)	Examiner:	Ellen C. Tran
		)		
Title:	Managing A Secure Environment Using A Chipset in Isolated Execution Mode	)		

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313 on:

6-23-2005
Date of Deposit

Gayle Bekish
Name of Person Mailing Correspondence

Gayle 6-23-2005
Signature Date

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

IN SUPPORT OF APPELLANT'S APPEAL

TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Sir:

Pursuant to Appellant's Notice of Appeal filed on even date herewith, Appellant hereby submits this Brief in support of its Appeal from the Final Office Action dated April 4, 2005 (hereinafter "the Final Office Action"). Appellant respectfully requests consideration of this Appeal by the Board of Patent Appeals and Interferences and allowance of the claims in the above-captioned patent application.

I. REAL PARTY IN INTEREST

The invention is assigned to Intel Corporation of 2200 Mission College Boulevard, Santa Clara, California 95052, as indicated by the assignment recorded at reel 011112, frame 0420.

II. RELATED APPEALS AND INTERFERENCES

To the best of Appellant's knowledge, there are no appeals or interferences related to the present appeal that will directly affect, be directly affected by, or have a bearing on the Board's decision.

III. STATUS OF CLAIMS

Claims 61-94 are pending in the application. Claims 1-60 were canceled before the Final Office Action. Claims 61-94 have been finally rejected. Claims 61, 72, and 84 are the independent claims. The rejections of all pending claims are appealed.

IV. STATUS OF AMENDMENTS

No amendments have been requested subsequent to the Final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Embodiments of the present invention relate to components that facilitate enhanced security and/or integrity for data processing systems. Those components include hardware within a processing system to support distinct modes of operation, referred to as "normal execution mode" and "isolated execution mode." (See, e.g., FIGs. 1A-1B; page 4, line 24 through page 5, line 24; claims 61, 72, 84.) The hardware also supports creation of "an isolated memory area" in the memory of the processing system, and the hardware ensures that the isolated memory area is "inaccessible from the normal execution mode." (See, e.g., FIG. 1B; page 6, lines 1-12; claims 61, 72, 84.)

In addition, the invention involves firmware (or other software) that executes as part of the process of booting up the processing system. In particular, a software component referred to as a "processor executive (PE) handler" is loaded into the isolated memory area during the process of booting up the processing system. (See, e.g., FIGs. 1A, 1C, 2; page 6, lines 8-20; page 12, line 22 through page 13, line 1; page 18, lines 18-24; claims 61, 72, 84.) The PE handler then manages "at least one subsequent operation in the boot process" from "the isolated execution mode." (See, e.g., page 5, lines 7-12; claims 61, 72, 84.) (As indicted on page 14, lines 19-22, the term "processor nub loader" refers to a particular implementation or embodiment of the broader term "PE handler.")

For reference, the example embodiment illustrated in FIG. 1A illustrates a processor nub loader/PE handler 52 to operate from isolated execution mode, and lines 10-11 on page 6 describe some boot operations to be managed by the processor nub loader/PE handler from the isolated execution mode, according to one example embodiment.

Furthermore, independent claim 61 and dependent claims 73 and 85 also specifically recite the operation of "obtaining at least part of the PE handler" from storage in a "chipset" of the processing system. Moreover, the other dependent claims recite numerous additional details. For instance, claim 76 recites the operation of a "storing a thread count" in the processing system, "the thread count indicating a number of threads operating in the isolated execution mode." Claim 78 recites the operation of "updating the thread count in response to access to [an] initialization storage" in the processing system. (E.g., see page 15, line 22 through page 17, line 1.)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether the Examiner erred in rejecting all claims (i.e., claims 61-94) under 35 U.S.C. § 102(e) as being anticipated by U.S. patent no. 6,226,749 to Marius M. Carloganu et al. (hereinafter "Carloganu").

B. Whether the Examiner erred in rejecting independent claim 61 and its dependent claims (i.e., claims 62-71), as well as dependent claims 73-74 and 85 under 35 U.S.C. § 102(e) as being anticipated by Carloganu.

C. Whether the Examiner erred in rejecting dependent claims 63-65, 76-78, and 87-89 under 35 U.S.C. § 102(e) as being anticipated by Carloganu.

VII. ARGUMENT

For a valid rejection under 35 U.S.C. § 102, "[t]he identical invention must be shown in as complete detail as is contained in the ... claim." (MPEP § 2131.01, quoting from Richardson v. Suzuki Motor Co., 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).)

Carloganu does not disclose all of the features recited in any of the pending claims

A. Claims 61-94 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Carloganu. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

(i) Regarding a processing system to support a normal execution mode and an isolated execution mode, and an isolated memory area to be inaccessible from the normal execution mode

As indicated above, each of independent claims 61, 72, and 84 refers to a processing system that supports "a normal execution mode" and "an isolated execution mode." Each of those claims also refers to memory in the processing system to include "an isolated memory area" that is "inaccessible from the normal execution mode."

By contrast, Carloganu disclose a processing system that includes an "application processing unit" and a separate "security module" (FIG. 2). The application processing unit seems to be a central processing unit (CPU), and the security module seems to be a security coprocessor residing in the same system with the CPU. In particular, Carloganu describes a particular approach for processing "secured commands" and "non-secured commands" in the security module, wherein the security module receives those commands from the application processing unit (col. 7, lines 33-42). For instance, an object of the Carloganu invention is "to provide a method and apparatus for operating a security module or other secure processor in which all of the application software program resides in an external application processing unit" (col. 2, lines 22-25).

Specifically, according to Carloganu, "an application program running in an external device" sends non-secured and secured commands to the secure processor for execution. The secure processor determines which commands are secured and which are non-secured by looking up each received command in a "command set up table." The secure processor then "immediately executes" the non-secured commands, and the secured processor only executes secured commands if those commands pass tests for "authenticity" and "regularity." (Abstract.)

Carloganu says nothing about supporting "isolated execution mode" and "normal execution mode." Carloganu also does not disclose memory to include an "isolated memory area" that is inaccessible from the normal execution mode.

With regard to claim 61, the Final Office Action asserts that column 10, lines 12-26 of Carloganu disclose an isolated memory area to be inaccessible from the normal execution mode. However, that portion of Carloganu discloses no such thing. Instead, that portion of Carloganu simply states that a "security module" can process "security module commands" when "the security of the system" has been "turned off." The cited lines say nothing about an "isolated memory area" within a memory of a processing system, wherein the isolated memory area is inaccessible from a processor operating in isolated execution mode.

For at least the foregoing reasons, Carloganu does not anticipate claims 61, 72, and 84, or any of the corresponding dependent claims.

<u>(ii) Regarding a PE handler to be loaded into the isolated memory area during a boot process and to manage at least one subsequent operation in the boot process from isolated execution mode</u>

Furthermore, claims 61, 72, and 84 each also involves a "PE handler" to be loaded into the isolated memory area during a boot process, with the PE handler to manage "at least one subsequent operation in the boot process" from the isolated execution mode.

As indicated above, Carloganu says nothing about supporting "isolated execution mode" and "normal execution mode," and Carloganu also does not disclose memory to include an "isolated memory area" that is inaccessible from the normal execution mode. A fortiori, Carloganu does not disclose loading a PE handler into an isolated memory area that is inaccessible from the normal execution mode.

With regard to claim 61, the Final Office Action asserts that column 7, lines 32-61 of Carloganu disclose a PE handler to be loaded into an isolated memory area. However, that portion of Carloganu discloses no such thing. Instead, that portion of Carloganu explains that the secure processor receives secured commands from the application processing unit, and the secure processor will execute "associated command primitives" if the command passes sequence and authenticity testing.

The cited lines also refer to "secure resources," indicating that the "term 'secured resource' is used to denote a resource under the control of the security module to distinguish from non-secured resources controlled by application processing unit 78. The term does not mean that the resources have their own physical or logical security features, though some may have such features."

Interestingly, the cited lines also specifically explain that "security is provided by rigorous sequence and authenticity testing of the secured commands. This feature of the invention prevents attackers from accessing the secured resources with Trojan horse programs that interrupt or replace commands in an authorized application software program running in application processing unit 60."

The cited lines say nothing about a "PE handler" to be loaded into an "isolated memory area" that is inaccessible from the normal execution mode. They also do not disclose a PE handler to manage "at least one subsequent operation in the boot process." They also do not disclose that the PE handler is to manage the at least one subsequent operation "from the isolated execution mode."

For at least the foregoing reasons, Carloganu does not anticipate claims 61, 72, and 84, or any of the corresponding dependent claims.

B. Claims 61-71, 73-74, and 85 stand finally rejected under 35 U.S.C. 102(e) as being anticipated by Carloganu. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

In addition to the features discussed above, claims 61-71, 73-74, and 85 each also involves storage in a chipset of the processing system, the storage to store at least part of the PE handler. As indicated above, the PE handler manages at least one subsequent operation in a boot process. Carloganu says nothing about obtaining any part of a PE handler from storage in a chipset.

With regard to claim 61, the Final Office Action asserts that column 7, lines 32-61 of Carloganu disclose storage in a chipset, the storage to store at least part of the PE handler. As indicated above, that portion of Carloganu explains that the secure processor receives secured commands from the application processing unit, and the secure processor will execute "associated command primitives" if the command passes sequence and authenticity testing. The cited lines also refer to "secure resources," indicating that the "term 'secured resource' is used to denote a resource under the control of the security module to distinguish from non-secured resources controlled by application processing unit 78. The term does not mean that the resources have their own physical or logical security features, though some may have such features." The cited lines also explain that "security is provided by rigorous sequence and authenticity testing of the secured commands."

The cited lines say nothing about storage in a chipset of the processing system, the storage to store at least part of the PE handler, wherein the PE handler manages at least one subsequent operation in a boot process.

For at least the foregoing reasons, Carloganu does not anticipate claims 61-71, 73-74, and 85.

C. Claims 63-65, 76-78, and 87-89 stand finally rejected under 35 U.S.C. 102(e) as being anticipated by Carloganu. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

As indicated above, claim 76 recites the operation of a "storing a thread count" in the processing system, "the thread count indicating a number of threads operating in the isolated execution mode." Claims 63 and 87 recite the same or similar features. In addition, claims 64-65 depend from claim 63, claims 77-78 depend from claim 76, and claims 88-89 depend from claim 87. Carloganu says nothing about storing a thread count in a processing system, wherein the thread count indicates "a number of threads operating in the isolated execution mode."

With regard to claim 63, the Final Office Action asserts that column 8, lines 40-52 of Carloganu disclose the operation of storing a thread count in a processing system, wherein the thread count indicates a number of threads operating in the isolated execution mode.

However, that portion of Carloganu discloses no such operation. Instead, that portion of Carloganu pertains to "an embodiment in which command sequence [i.e., the sequence of commands to be executed by the secure processor] is required to be sequential." Specifically, the Carloganu states that secured commands can have a "format including a command sequence ID, a command code, and a set of command data items" (col. 8, lines 34-38.) Before executing a command, the secured processor may check to make sure that the command has the correct sequence ID. The cited lines say nothing about storing any kind of thread count, let alone a thread count indicating a number of threads operating in the isolated execution mode.

For at least the foregoing reasons, Carloganu does not anticipate claims 63-65, 76-78, and 87-89.

In addition, as indicated above, claim 78 recites the operation of "updating the thread count in response to access to [an] initialization storage" in the processing system. Claims 64 and 89 recite the same or similar features.

With regard to claim 64, the Final Office Action asserts that column 8, lines 16-67 of Carloganu disclose the operation of updating a thread count in response to access to an initialization storage. However, that portion of Carloganu discloses no such operation.

Instead, as indicated above, that portion of Carloganu states that secured commands can have a "format including a command sequence ID, a command code, and a set of command data items;" and before executing a command, the secured processor may check to make sure that the command has the correct sequence ID. The cited lines do not disclose the operation of updating a thread count in response to access to an initialization storage.

For at least the foregoing reasons, Carloganu does not anticipate claims 64, 78, and 89.

Information Disclosure Statements

Appellant has submitted numerous Information Disclosure Statements (IDSs) and electronic IDSs (eIDSs) for this patent application, and the Examiner has provided confirmation of consideration for most of those submissions. However, Appellant has not received confirmation of consideration for the following three documents: the IDS that was submitted on September 21, 2004; and the two eIDSs that were submitted on November 8, 2004. Appellant therefore respectfully requests confirmation that the Examiner has considered all of the references listed in those three documents.

Conclusion

Appellant respectfully submits that all pending claims in this patent application are patentable, and requests that the Board of Patent Appeals and Interferences overrule the Examiner and direct allowance of the rejected claims. Appellant also respectfully requests confirmation that the Examiner has considered the IDS and eIDSs referenced above.

If any fee insufficiency or overpayment is found, please charge any insufficiency or credit any overpayment to Deposit Account No. 02-2666.

Respectfully submitted,

Intel Corporation

Date:____June 23, 2005______ ____/ Michael R. Barre /________________

Michael R. Barré
Registration No. 44,023
Patent Attorney
Intel Americas, Inc.

Attorney Phone Number: (512) 732-3923

Correspondence Address: Blakely Sokoloff Taylor & Zafman, LLP
12400 Wilshire Blvd
Seventh Floor
Los Angeles, California 90025-1026

VIII. CLAIMS APPENDIX

1-60. (canceled)

61. (previously presented) A processing system comprising:

a processor to support an isolated execution mode, a normal execution mode, a first privilege ring within the normal execution mode for use by an operating system (OS) kernel, and a second privilege ring within the normal execution mode for use by a software application;

a memory responsive to the processor, the memory to include an isolated memory area, the isolated memory area to be inaccessible to the processor in the normal execution mode;

a chipset responsive to the processor, the chipset to support the normal execution mode and the isolated execution mode;

processor executive (PE) handler storage in the chipset to store at least part of a PE handler, the PE handler to be loaded into the isolated memory area during a boot process for the processing system after at least a portion of the processing system is initialized, the PE handler to manage, from the isolated execution mode, at least one subsequent operation in the boot process.

62. (previously presented) The processing system of claim 61, wherein the processing system enters the isolated execution mode before loading the PE handler into the isolated memory area.

63. (previously presented) The processing system of claim 61, further comprising:

a thread count storage, the processing system to store, in the thread count storage, a thread count indicating a number of threads operating in the isolated execution mode.

64. (previously presented) The processing system of claim 63, further comprising:

an initialization storage, the processing system to update the thread count in response to access to the initialization storage.

65. (previously presented) The processing system of claim 63, wherein the processing system provides indication of a failure mode in response to the thread count reaching a thread limit.

66. (previously presented) The processing system of claim 61, further comprising:

a log storage to store identifiers of executive entities operating in the isolated execution mode.

67. (previously presented) The processing system of claim 61, further comprising:

key storage to store a key to be used to handle one or more executive entities to operate in the isolated execution mode.

68. (previously presented) The processing system of claim 67, wherein the key comprises data based on a substantially random value generated by a manufacture of hardware for the processing system.

69. (previously presented) The processing system of claim 61, further comprising:

storage responsive to the processor; and

at least one executive entity encoded in the storage, the at least one executive entity selected from the group consisting of a processor executive (PE) and an operating system executive (OSE), the at least one executive entity to operate in the isolated execution mode.

70. (previously presented) The processing system of claim 61, further comprising:

configuration storage to store a base value and a mask value, the processing system to establish the isolated memory area in the memory based at least in part on the base value and the mask value.

71. (previously presented) The processing system of claim 61, wherein the PE handler storage comprises substantially non-volatile storage.

72. (previously presented) A method comprising:

initializing a processing system during a boot process for the processing system, wherein the processing system comprises a processor and a memory, the processing system to support an isolated execution mode, a normal execution mode, a first privilege ring within the normal execution mode for use by an operating system (OS) kernel, and a second privilege ring within the normal execution mode for use by a software application;

during the boot process, establishing an isolated memory area in the memory, the isolated memory area to be inaccessible from the normal execution mode; and

after at least a portion of the processing system is initialized, loading a processor executive (PE) handler into the isolated memory area, the PE handler to manage, from the isolated execution mode, at least one subsequent operation in the boot process.

73. (previously presented) The method of claim 72, wherein the processing system further comprises a chipset with a PE handler storage, the method further comprising:

obtaining at least part of the PE handler from the PE handler storage of the chipset.

74. (previously presented) The method of claim 73, wherein the PE handler storage comprises substantially non-volatile storage.

75. (previously presented) The method of claim 72, further comprising:

entering the isolated execution mode before loading the PE handler into the isolated memory area.

76. (previously presented) The method of claim 72, wherein the processing system further comprises a thread count storage, the method further comprising:

storing a thread count in the thread count storage, the thread count indicating a number of threads operating in the isolated execution mode.

77. (previously presented) The method of claim 76, further comprising:

providing indication of a failure mode in response to the thread count reaching a thread limit.

78. (previously presented) The method of claim 76, wherein the processing system further comprises an initialization storage, the method further comprising:

updating the thread count in response to access to the initialization storage.

79. (previously presented) The method of claim 72, further comprising:

operating one or more executive entities in the isolated execution mode; and

storing identifiers of the executive entities operating in the isolated execution mode.

80. (previously presented) The method of claim 72, wherein the processing system comprises key storage to store a key, the method further comprising:

using the key to handle one or more executive entities to operate in the isolated execution mode.

81. (previously presented) The method of claim 80, wherein the key comprises data based on a substantially random value generated by a manufacture of hardware for the processing system.

82. (previously presented) The method of claim 72, further comprising

operating one or more executive entities in the isolated execution mode, wherein the executive entities comprise at least one entity selected from the group consisting of a processor executive (PE) and an operating system executive (OSE).

83. (previously presented) The method of claim 72, wherein the processing system comprises configuration storage to store a base value and a mask value, and the operation of establishing an isolated memory area in the memory comprises:

using the base value and the mask value to establish the isolated memory area.

84. (previously presented) An apparatus comprising:

a machine accessible medium; and

instructions encoded in the machine accessible medium, wherein the instructions, when executed by a processor of a processing system, perform operations comprising:

initializing at least part of the processing system during a boot process for the processing system, the processing system to support an isolated execution mode, a normal execution mode, a first privilege ring within the normal execution mode for use by an operating system (OS) kernel, and a second privilege ring within the normal execution mode for use by a software application;

during the boot process, establishing an isolated memory area in a memory of the processing system, the isolated memory area to be inaccessible from the normal execution mode; and

after at least a portion of the processing system is initialized, loading a processor executive (PE) handler into the isolated memory area, the PE handler to manage, from the isolated execution mode, at least one subsequent operation in the boot process.

85. (previously presented) The apparatus of claim 84, wherein the processing system further comprises a chipset with a PE handler storage, and the instructions perform operations comprising:

obtaining at least part of the PE handler from the PE handler storage of the chipset.

86. (previously presented) The apparatus of claim 84, wherein the instructions perform operations comprising:

causing the processor to enter the isolated execution mode before loading the PE handler into the isolated memory area.

87. (previously presented) The apparatus of claim 84, wherein the processing system further comprises a thread count storage, and the instructions perform operations comprising:

storing a thread count in the thread count storage, the thread count indicating a number of threads operating in the isolated execution mode.

88. (previously presented) The apparatus of claim 87, wherein the instructions perform operations comprising:

providing indication of a failure mode in response to the thread count reaching a thread limit.

89. (previously presented) The apparatus of claim 87, wherein the processing system further comprises an initialization storage, and the instructions perform operations comprising:

updating the thread count in response to access to the initialization storage.

90. (previously presented) The apparatus of claim 84, wherein the instructions perform operations comprising:

causing one or more executive entities to operate in the isolated execution mode; and

storing identifiers of the executive entities operating in the isolated execution mode.

91. (previously presented) The apparatus of claim 84, wherein the processing system comprises key storage to store a key, and the instructions perform operations comprising:

using the key to handle one or more executive entities to operate in the isolated execution mode.

92. (previously presented) The apparatus of claim 91, wherein the key comprises data based on a substantially random value generated by a manufacture of hardware for the processing system.

93. (previously presented) The apparatus of claim 84, wherein the instructions perform operations comprising:

causing one or more executive entities to operate in the isolated execution mode, wherein the executive entities comprise at least one entity selected from the group consisting of a processor executive (PE) and an operating system executive (OSE).

94. (previously presented) The apparatus of claim 84, wherein the processing system comprises configuration storage to store a base value and a mask value, and the operation of establishing an isolated memory area in the memory comprises:

using the base value and the mask value to establish the isolated memory area.

IX. EVIDENCE APPENDIX

Not Applicable

X. RELATED PROCEEDINGS APPENDIX

Not Applicable